National University of Computer and Emerging Sciences

Lab Manual 02
Object Oriented Programming

Course Instructor	Ms. Syeda Tayyaba Bukhari
Lab Instructor (s)	Fariha Maqbool Muhammad Usama
Section	BDS-2B
Semester	Spring 2022

Department of Computer Science
FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

- ✓ Dynamically allocate and deallocate memory
- ✓ Access dynamically allocated memory via subscript and offset notation
- ✓ Pass a pointer to function
- ✓ Return pointer from a function

TASK 1:

Write a C++ program that declares and initializes a float array dynamically and finds the index of the first occurrence of the second largest element in the array.

For Example:

Input:
Please enter size: 5
Please enter elements: 1.5
7.8
3.2
9.0
7.1

Output:
Second Largest element is: 7.8
Index of second largest element is: 1

TASK 2:

Fibonacci sequence is a sequence in which every number after the first two is the sum of the two preceding ones.

a) Write a C++ program that takes a number n from user and populate a dynamic array with first n Fibonacci numbers.

For example:

Input:
Please enter size: 10

Output:
Fibonacci Numbers: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55

b) Modify your code such that the output is now printed via a function.
void Output (int * myptr, int size)

TASK 3:

Write a program that keeps taking integer input from the user until user enters -1 and displays the data in reverse order.
Your program should save the input in a dynamically allocated array. Initially create a dynamic array of five integers. Each time the array gets filled your program should double the size of array (i.e. create a new array of double size, copy previous data in new array, delete previous array) and continue taking the input. After receiving -1 (i.e. end of data input) your program should print the numbers in the reverse order as entered by the user.

Important Note: subscript operator [] is not allowed to traverse the array. Use only offset notation. i.e instead of using myArray[i] use *(myArray+i) to read/write an element. Do not consume extra space. There shouldn't be any memory leakage or dangling pointers in your code.

You have to make use of the following functions for this task:

- void Input (int * & iarr, int & size); //why is size passed by reference for this?
- void reverse (int * iarr, int size);
- void Output (int * iarr, int size);

TASK 4:

Take size input from the user and create an array of that size. Now populate the array as well by taking input from the user.

- First Implement **void copyArray(int* arr, int *&arr1, int size)** that copies arr into arr1.
- Now implement another function **int reduceArray(int *arr, int *&arr1, int size)** that asks user to enter size to reduce the array. To reduce the array remove (or chop) the elements of the arr from the start and copy remaining into arr1. Use copyArray function to copy.

For Example:

Input:
Please enter size: 7
Please enter elements: 91
3
40
7
8
12
642
Please enter the reduced size of array: 5

Output:
Array after reduction is: 40, 7, 8, 12, 642

TASK 5:

Write a program that takes size of an array and its elements and removes consecutive occurrences of same number from the list. For the example given below, your program should have space of exactly 7 integers on heap after compression. Do not consume any extra byte on heap.

For Example:

Input:
Please enter size: 18
Please enter elements: 1,1,2,2,2,3,4,5,5,5,5,5,7,7,7,2,2,2

Output:
Array After Compression: 1,2,3,4,5,7,2

Use the following main() for this question:

```
int main()
{
	int size;
	cout<<"Enter the size of array"<<endl;
	cin>>size;
	int* c1 = new int [size];

	cout<<"Input elements in array: ";
	Input(c1, size);

	//Compressing the array
	int * p2=Compression (c1, size);

	cout<<"Array after compression will be: ";
	Output(p2, size);

	system("pause");
	return 0;
}
```